IFT6132 - 2D human pose estimation using a hybrid CNN and Markov Random Field model

David Abraham [1] **Simon Dufort-Labbé** [1]

Abstract

This paper presents a Pytorch implementation of the paper "Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation" (Tompson et al., 2014). The model is a hybrid architecture of a CNN and a Markov Random Field that is applied to the problem of 2D pose estimation using a single image. We show that this model can be trained jointly in a single step instead of the longer and more complicated three steps process proposed in the paper.

1. Introduction

Human pose estimation from a single 2D image is a very interesting and important problem that, while having received a huge amount of attention, still remains a challenging task due to many factors: complex joint dependency, joint occlusions, variation in body parts shape and length, different lightings in images, etc.

The goal of this project is to perform human pose estimation using an approach based on the paper "Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation" (Tompson et al., 2014). The paper uses a model that combines a body part detector, which is a fully convolutional network, to predict a heatmap of body part locations and a graphical high level spatial model to refine the output of the CNN by enforcing kinematics constraints to the body part locations.

The use of a spatial model to refine the output of another model has already been done, but in these previous works the graph structure of the spatial model was hand crafted. The biggest contribution of this paper is that the body part detector and the spatial model can jointly be trained so that no hand crafted features are used.

2. Dataset

The dataset used in this project is the Frames Labeled In Cinema (FLIC) dataset (Sapp & Taskar, 2013). The dataset contists of 5003 images taken from Hollywood movies. The resolution of the images is 720x480 pixels with 3 channels (colored). Each image is labeled with body parts joint location: left and right wrists, left and right elbow, left and right shoulder, left and right hips, nose and torso. 80% of data is used for training (3987 images) and 20% is used for testing (1016 images). In the original dataset, left and right eyes are also annotated, but they are not used in our project as we are not interested in such precise information.

The location of each joint (body part) is given by a heatmap with a resolution of 90x60 pixels with a single channel. Since there are 10 joints, for each image we have a heatmap of 90x60 pixels with 10 channels (one channel per body part). The heatmap value is 0 for every pixel except where the body part is located. There, a small gaussian is used instead of a single 1 to account for the incertitude of the body part location.

Since we had a time constraint for handing in the project, we used a subset of the dataset so that the training process is faster. We used a random subset of 2500 training images and 1000 testing images.

3. Model description

3.1. Body part detector

The paper proposes to use a CNN for the body part detector. Figure 1 shows the final architecture they use in their implementation.

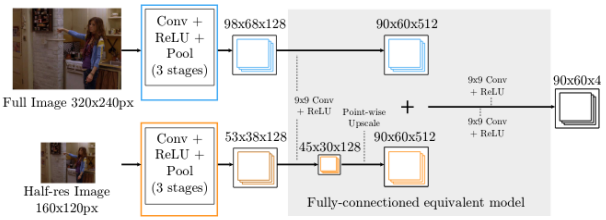


Figure 1. Body part detector architecture used in the paper. (Tompson et al., 2014)

They use a fully convolutional model, so there is no fully connected layers in the model. As input to the network, they use two images of different resolutions to capture a wider

range of spatial context. Both input images are processed by multiple convolution-relu-max pool layers to generate feature maps of smaller resolution : 98x68 pixels for the full image and 53x38 pixels for the half-res image, both with 128 channels. These feature maps are then processed further by a convolution-relu layer with a large 9x9 kernel size. Both feature maps are then scaled to 90x60 pixels (upscaled in the case of the half-res input) with 512 channels, added together and processed further by two convolution-relu layers with the same large 9x9 kernel size to produce the final output. The output is a 90x60 pixels heatmap with a channel for each body part (e.g. left hip, right shoulder, etc.) location that the model predicts.

The part detector described here takes as input a 320x240 pixels image which they state is the "full image" and an image of size 160x120 pixels which they state is at "half-res". The images in the FLIC dataset are 720x480 pixels and there is no justification anywhere in the paper for this precise resizing to a 320x240 resolution (which is not even exactly half the original resolution) and its use as input instead of the original resolution. In their model, it is not clear how they go from 320x240 pixels to 98x68 pixels feature maps after 3 stages of convolutional layers. In our model, we used a more natural approach, starting from the full resolution image (720x480) without any special preprocessing applied. We also use three different rescaling of the input image instead of two: one at full resolution (720x480), one at half resolution (360x240) and one at quarter resolution (180x120). This allows us to reconcile the direct use of the full resolution image as input while capturing the same spatial context as the model described in the original paper.

The training of the model is done in a supervised manner using Adam, as opposed to SGD in the paper, as it is known to work very well with default parameters. In the paper they state that they use a MSE loss for training the CNN, but in our implementation we use instead a softmax activation separately on each outputted heatmap with a binary cross entropy loss, as this seems like a better choice, to minimize the distance between the predicted output heatmap and the target heatmap for each body part.

3.2. Spatial Model

The role of the spatial model is to leverage the intrinsic structure between joints to perfect the prediction made by the part detector model. The idea behind it is simple : knowing the position of the left shoulder, you should be able to make a general guess about the position of the right shoulder.

To model this and to make the structure learnable, the author suggests the use of a Markov Random Field (MRF) like model. To make it quite generic, a fully connected graph is made between all joints. The final probability density of joint A over the pixel map can therefore be represented as a product of potentials such as:

$$\begin{aligned} P_{\bar{A}} &= \frac{1}{Z_1} \prod_{v \in \mathcal{V}} \phi_{A,v}(x_{A,v}) * \phi_v(x_v) \\ &= \frac{1}{Z_2} \prod_{v \in \mathcal{V}} p_{A|v} * p_v + b_{v \to A} \end{aligned} \tag{1}$$

$\mathcal{V}$ being the set of all joints and where p_v is taken to be the output of the part-detector model, $p_{A|v}$ are learned parameters : the quantity $p_{A|v} * p_v$ being represented by a convolution between the map $p_{A|v}$ and the part detector output p_v, thus effectively allowing the model to learn the position of the joint A with regard to joint v.

The parameters $b_{v \to A}$, which are biases, are also learned and represent the base probability of getting a message from v to A for every pixel location. This term is essential, in the case where p_v and $p_{A|v}$ would be disjoint; therefore ensuring that $P_{\bar{A}}$ do not fall to zero at locations where a detection of the investigated joint was made by the part-detector. Moreover, the term $b_{A \to A}$ allows a better fit of the model to the training distribution : for example in the dataset used here, some images contain more than one person but there is always only one person whom joints are tagged. On such cases, the image is usually centered around the tagged subject; the term $b_{A \to A}$ allow the model to learn this particular bias.

Finally, let's note that the author proposes to treat the spatial model as an energy model, and thus suggests the following estimation to equation (1) to facilitate its training :

$$\begin{aligned} \bar{e}_A = \exp\Bigg(\sum_{v \in \mathcal{V}} \big[&\log\big(\mathrm{SoftPlus}(e_{A|v}) * \mathrm{ReLU}(e_v) \\ &+ \mathrm{SoftPlus}(b_{v \to A})\big) \big] \Bigg) \\ \text{where :} \quad \mathrm{SoftPlus}(x) &= \frac{1}{\beta} \log(q + \exp(\beta x)) \end{aligned} \tag{2}$$

The use of an energy model simplifies the overall expression, not having to care about the normalization constant anymore.

4. Differences with the paper

The article was pretty hard to reproduce exactly since they do not give all details for their model architecture and their hyperparameters. They also do not have their implementation available (for example on a github repository). This meant that we had to take some liberties concerning the details of the model architecture and the training procedure.

This section will outline the main modifications or choices we made since there was no official baseline.

4.1. Architecture

4.1.1. PART-DETECTOR

The main difference we brought to the part-detector was to generate the different image scale using a Features Pyramidal Network like (Lin et al., 2016) approach instead of resizing the image and treat the different scales as different input. Instead, we used a learned convolutional layer with a stride of two to rescale the image before sending each generated scale to a different branch (no parameters sharing between branches). When upscaling was needed, it was also done in a similar fashion.

Another difference is that we added batch normalisation layers to our model to improve the speed and stability of the learning process.

4.1.2. SPATIAL MODEL

For the spatial model, we mostly respected the structure proposed by the authors, filling the gap with our own understanding when needed. A few variations were made for consistency and stability reason during the training. The following expression summarizes well the operations performed by our implementation of the spatial model:

$$\bar{p}_A \approx \text{Softmax}\left(\sum_{v\in\mathcal{V}}\left[\log\left(\text{Conv2d}\left[\text{SoftPlus}(p_{A|v}) \ * \ \text{SoftPlus}(p_v)\right] + \text{SoftPlus}(b_{v\to A})\right)\right]\right) \tag{3}$$

The use of the $Conv2d$ operation, hinted by the author, really allows to take profit of the GPU acceleration during training. Moreover, it is totally suitable as the parameters $p_{A|v}$ can be seen as a map indicating the likely position of joint A with regard to a centered joint v

Another big difference with the original paper is that we decided to apply normalization to the output produced by the spatial model. Their argument for not doing it was that it allowed for the detection of multiples joints of the same type (in the case where multiple persons are present on the evaluated image). Contrary to them, we found that normalizing gave neater results, with less indecision on the joints position. We think that the difference in quality stem directly from the dataset used for training, and this for two reasons:

1. Even when multiple persons are present in the training image, only one has his joints tagged, and it will either be the more centered person or the person in the foreground.

2. For the vast majority of the training examples, all possible joints are visible on the image. By normalizing during training, we allow the model to learn this bias toward this subject selection (the person whom the model will predict the joints). Learning this bias seems to improve the precision for the final prediction. This behaviour is exposed in the results section (Figure 3).

Still we need to remark that the authors variant could be more appropriate on a different dataset.

4.2. Training procedure

In the paper, they used a three stages training process. They first train the body part detector and the spatial model separately, than they finally fine tune the two together after merging them into an end-to-end fashion. Moreover, they initialize the "likelihood" ($p_{A|v}$) portion of the spatial model by making a 2D histogram over the training dataset of the position of other joints with regard to the studied joint.

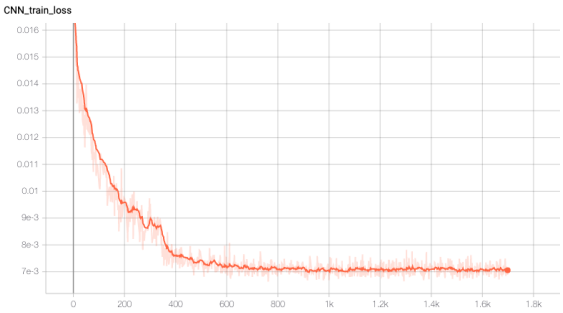
Instead, we use a single joint training process of the whole model, which is faster, simpler and turned out to be able to achieve the same kind of results; probably due to the better hardware and proven optimization schemes available nowadays. For this, the "likelihood" weights were initialized using a uniform distribution, and the network learned the correct distribution with regard to the data. All in all, it is a more representational learning approach.

5. Results

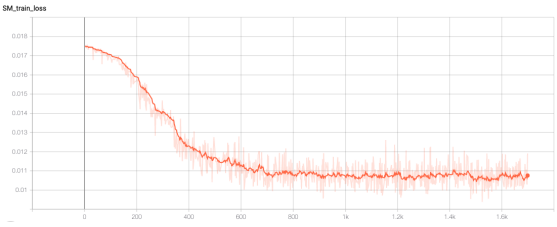
The models described in section 3.1 and 3.2 were implemented using the Pytorch library. The training of both models in a joint fashion from scratch took around 1-2 hours on a GPU. The complete code of our implementation can be found on our github repo (`www.github.com/dvidbruhm/INF6132-cnn-mrf-pose-estimation`).

Here what we are mostly interested is to do a comparison between results from the part detector model to the version incorporating the spatial model on top of it. Let's note that the incorporation of the spatial model is what allows the network to improve its structured predictions.

We can begin this comparison by looking at the training procedure. On the figure 2 below, we display the training behaviour. First, the model perfects its part detector until this stage is making good enough predictions. Then, the spatial model really starts to improve, learning the underlying distribution between joints location.



. a) Loss of CNN body part detector during training.



. b) Spatial model loss during training.

Figure 2. The learning behaviour of the cnn body part detector (a), and of the final model (b), displayed according to a cross-entropy loss when compare to the target images

Figure 3 compares the prediction of the CNN body part detector to the prediction of the complete model that includes the spatial model. We can see in figure 3a) that the predictions of the CNN body part detector alone can already find the location of the joints accurately on the person in the scene, but there still are a lot of false positives visible from the colored output scattered in the scene all around. Figure 3b) shows this prediction refined by the spatial model. We can clearly see that the spatial model does a tremendous job at removing almost all joint predictions that are not kinematically acceptable, i.e. the false positives detected by the CNN. Visually, no false positives remain. Figure 3c) shows the target joint locations. When comparing figures 3b) and 3c) we can clearly see that our model predictions are very accurate.

Figure 4 shows an example output of the prediction of the left shoulder location based on the other joints location: a) based on the left shoulder, b) based on the left elbow and c) based on the face. Note that there is a similar heatmap to those displayed for each of the 10 joints. In these three examples, we can see that all the heatmaps have a maximum prediction mostly around the left shoulder. By summing up all those heatmaps and the CNN output, we can find an accurate location for the left shoulder that respects the body constraints for all the joints while rejecting all false positives in the CNN output.

6. Conclusion

We have successfully implemented the paper "Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation" (Tompson et al., 2014) by showing that a joint model consisting of a CNN body part detector and a graphical spatial model to refine the output of the CNN gives very good results. We improve the training method of the paper that consists of a three steps procedure by simplifying it to training the whole joint model in a single step.

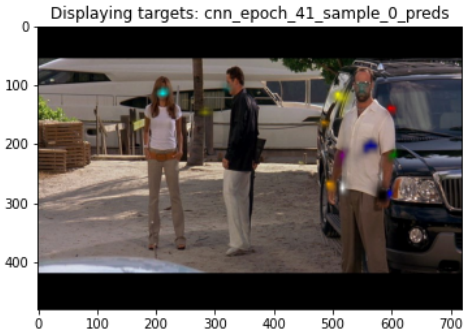
Since the paper was released in 2014, more modern CNN architectures such as resnet and VGG where not yet created. We can imagine that using such architectures for the CNN body part detector would greatly improve the performance of the model.

References

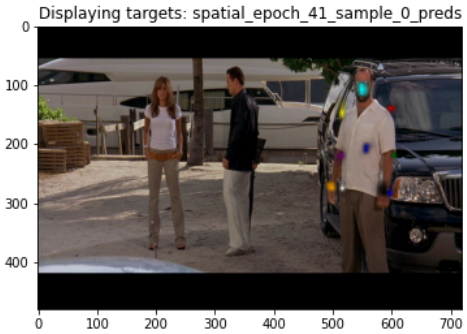
Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. Feature pyramid networks for object detection, 2016.

Sapp, B. and Taskar, B. Modec: Multimodal decomposable models for human pose estimation. pp. 3674–3681, 06 2013. doi: 10.1109/CVPR.2013.471.

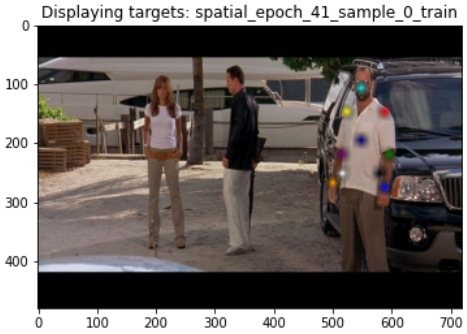
Tompson, J., Jain, A., LeCun, Y., and Bregler, C. Joint training of a convolutional network and a graphical model for human pose estimation, 2014.



. a) Output prediction of the CNN body part detector.

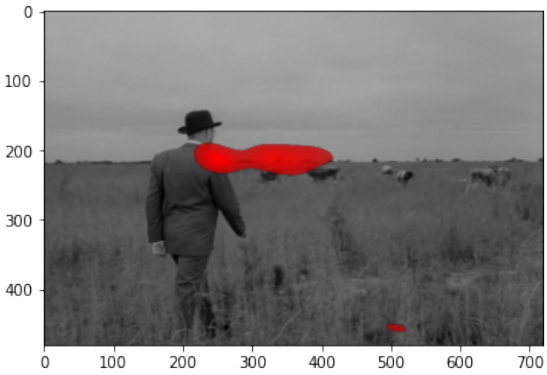


. b) Output prediction of the final model.

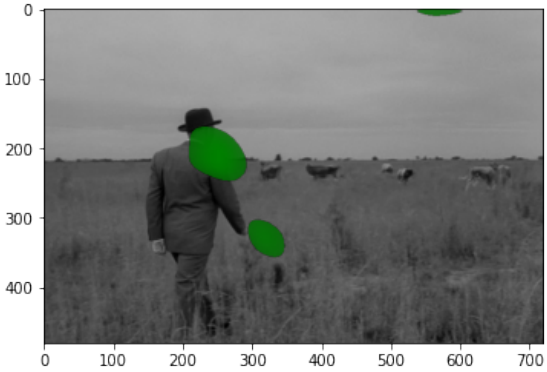


. c) Target joint locations.

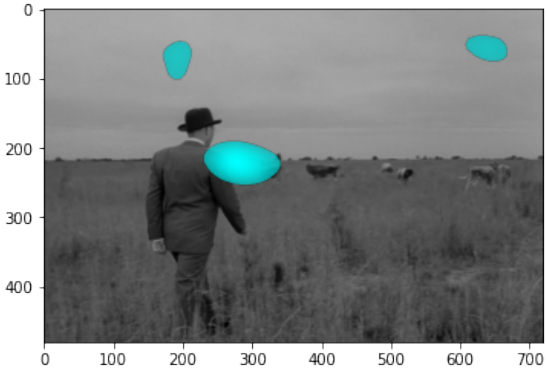
Figure 3. Example output of the trained model prediction for all joints of the cnn body part detector (a), the final model (b) and the target values (c).



. a) Prediction of left shoulder based on right shoulder.



. b) Prediction of left shoudler based on left elbow.



. c) Prediction of left shoulder based on the face.

Figure 4. Example output of the prediction of the left shoulder location based on several other joints: a) based on the right shoulder, b) based on the left elbow and c) based on the face.

A. Additional detection examples

Figure 5. On the left are examples that have only been processed by the part-detector, while the images on the right have also gone through the spatial model.